Toward Never Ending Language Learning for Morphologically Rich Languages

Buraya, Kseniya

The Association for Computational Linguistics
2017

Buraya , K , Pivovarova , L , Budkov , S & Filchenkov , A 2017 , Toward Never Ending Language Learning for Morphologically Rich Languages . in Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing . The Association for Computational Linguistics , Stroudsburg, PA , pp. 108-118 , Workshop on Balto-Slavic Natural Language Processing , Valencia , Spain , 04/04/2017 . https://doi.org/10.18653/v1/w17-1417

http://hdl.handle.net/10138/214844
https://doi.org/10.18653/v1/w17-1417

Toward Never Ending Language Learning for Morphologically Rich Languages

Kseniya Buraya
ITMO University, Russia
ksburaya@corp.ifmo.ru

Lidia Pivovarova
University of Helsinki, Finland
pivovaro@cs.helsinki.fi

Sergey Budkov
ITMO University, Russia
s.a.budkov@gmail.com

Andrey Filchenkov
ITMO University, Russia
afilchenkov@corp.ifmo.ru

Abstract

This work deals with ontology learning from unstructured Russian text. We implement one of the components of Never Ending Language Learner and introduce the algorithm extensions aimed to gather specificity of morphologically rich free-word-order language. We perform several experiments comparing different settings of the training process. We demonstrate that morphological features significantly improve the system precision while seed patterns help to improve the coverage.

1 Introduction

Nowadays a big interest is paid to systems that can extract facts from the Internet (Pasca et al., 2006; Choo et al., 2013; Grozin et al., 2016; Dumais et al., 2016; Samborskii et al., 2016).

The main challenge is to design systems that do not require any human involvement and may efficiently store lots of information limited only by the amount of the knowledge uploaded to the Internet. One of the ways of representing information for such systems is *ontologies*.

According to the famous definition by Gruber (1995), ontology is "an explicit specification of a conceptualization", i.e. formalization of knowledge that underlines language utterance. In the simplest case, ontology is a structure containing *concepts* and *relations* among them. In addition, it may contain a set of axioms that define the relations and constraints on their interpretation (Guarino, 1998). One of the advantages of such structures is data formalization that simplifies the automatic processing. Ontologies are widely used in information retrieval, texts analysis and semantic applications (Albertsen and Blomqvist, 2007; Staab and Studer, 2013).

In many practical applications, ontological concepts should be associated with *lexicon* (Hirst, 2009), i.e. with language expressions and structures. Even though ontologies themselves contain knowledge about the world, not a language, their primary goal is to ensure semantic interpretation of texts. Thus, *ontology learning* from text is an emerging research direction (Maedche, 2012; Staab and Studer, 2013).

One of the approaches that are used to learn facts from unstructured text is called *Never Ending Language Learning* (NELL) (Carlson et al., 2010a).[1] One of the NELL advantages is its low demand for preprocessed data required for the learning process. Given an initial ontology that contains 10–20 seeds for each category as an input, NELL can achieve a high performance level on extracting facts and relations from a large corpus (Carlson et al., 2010a).[2]

The first implementation of NELL (Carlson et al., 2010a) worked with English. An attempt was made to extend the NELL approach for the Portuguese language (Duarte and Hruschka, 2014). The main result of these experiments was that applying initial NELL parameters and ontology to non-English web-pages would not show high results; initial configuration did not work well with Portuguese web-pages. The authors made a conclusion that in order to extend the NELL approach to a new language, it is necessary to prepare a new seed ontology and contextual patterns that depend on the language rules.

In this paper, we introduce a NELL extension

[1] In this paper, we will use term "NELL" to refer both the approach and its implementations since it is traditional for the corresponding papers and the project.

[2] We distinguish two types of concepts: *categories* that are top-level concepts in predefined ontology and *instances*, that are descendants of top-level concepts; instances, apart from small initial seeds, are learned from text.

to the Russian language. Being a Slavic language, Russian has a rich morphology and free word order. Thus, common expressions for semantic relations in text have a specific form: the word order is less reliable than for Germanic or Romance languages; the morphological properties of words are more crucial. However, many pattern learning techniques are based on word order of pattern components and usually do not include morphology. Thus, the adaptation of the NELL approach to a Slavic language would require changes in the pattern structure. We introduce an adaptation of NELL to Russian, test it on a small dataset of 2.5 million words for 9 ontology categories and demonstrate that utilizing of morphology is crucial for ontology learning for Russian. This is the main contribution of this paper.

The rest of the paper is organized as follows. Section 2 overviews original NELL approach. Our improvements of the algorithm are presented in Section 3. Section 4 describes our data source, its preprocessing, and experiments we run. Results of these experiments are presented and discussed in Section 5. In Section 6, we give a brief overview of the related papers. We summarize the results and outline the future work in Section 7.

2 Never Ending Language Learner

The NELL architecture, which is presented in Figure 1, consists of two major parts: a knowledge base (KB) and a set of iterative learners (shown in the lowest part of the figure). The system works iteratively: first, the learners try to extract as much candidate facts as possible given a current state of the KB; after that, the KB is updated using learners output. This process is running infinitely, with the current state of KB being freely available at the project webpage.[3]

In this work, we focus on one of the NELL components, namely Coupled Pattern Learner (CPL). CPL is the free-text extractor that learns contextual patterns to extract instances of ontology categories. The key idea of CPL is that simultaneous ("coupled") learning of instances and patterns yields a higher performance than learning them independently (Carlson et al., 2010b).

An expression that matches text in CPL consists of three parts, which must be found within the same sentence:

[3] http://rtw.ml.cmu.edu/rtw/

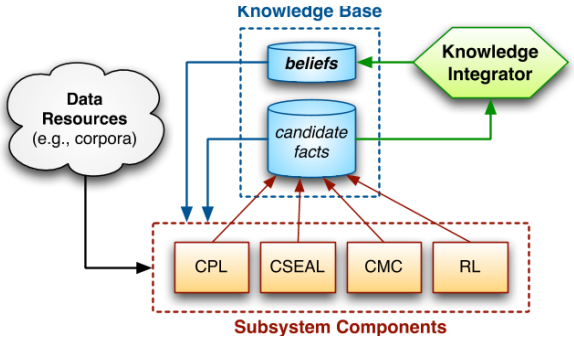

Figure 1: NELL architecture adapted from (Carlson et al., 2010a).

1. Category word. The list of category words is fixed and defined in the initial ontology.

2. Instance extracting pattern. A pattern consists of at most three words including punctuation like commas or parenthesis, but excluding category and instance words.

3. Instance word. At the beginning 3–5 seed instances are defined for each category.

CPL uses two sets: the set of *trusted patterns* and set of *trusted instances*, which are considered to be actual patterns and instances for the corresponding category. Different implementations may or may not exclude patterns/instances from the corresponding sets during further iterations.

The process starts with a text corpus and a small seed ontology that contains sets of trusted patterns and trusted instances. Then every learning iteration consists of the two following steps:

- **Instance extraction**. To extract new instances, the system finds a co-occurrence of the category word with a pattern from the trusted list and then identify the instance word. If both category and instance words satisfy the conditions of the pattern, then the found word is added to the pool of candidate instances for the current iteration. When all sentences are processed, candidate instance evaluation begins after which the most reliable instances are added to the set of trusted instances;

- **Pattern extraction**. To extract new patterns, the system finds a co-occurrence of the category word with one of its trusted instances. The sequence of words between category and instance are identified as a candidate pattern.

When all candidate patterns are collected, the most reliable patterns are added to the trusted set.

3 The Proposed Approach

3.1 Adaptation to the Russian Language

Russian patterns should have a specific structure, which should comprise morphological components. Thus we expand the form of the search expression so that case and number are taken into account for both category and instance words.

Let us consider an example, which illustrates importance of including morphology into patterns:

Тренеры знают множество приемов для дрессировки собак, такие как поощрение едой и многие другие.

Coaches know many techniques for training dogs, such as stimulation with food and etc.

This sentence matches *such as* pattern and without morphological constraints that may lead to extracting of wrong relations "**stimulation** *is a* **dog**". If the pattern have specified only part-of-speech rules, then our algorithm would produce a lot of errors. Specification of the arguments (nominative in this example) helps to avoid such false pattern triggering. Another way to avoid such errors would be a syntax annotation of all data and running CPL on top of this annotation; we leave this approach for further research.[4]

3.2 Strategies for Expanding the Trusted Sets

To add new patterns and instances to the corresponding trusted sets, we use *Support* metric. For each category, instances and patterns are ranked separately using the following formulas:

$$Support_c^{(t)}(i) = \frac{\sum_{p \in \text{TruPat}_c^{(t-1)}} Count_c(i,p)}{Count_c(i)}$$

for instances and

$$Support_c^{(t)}(p) = \frac{\sum_{i \in \text{TruInst}_c^{(t-1)}} Count_c(i,p)}{Count_c(p)}$$

for patterns, where i is an instance word, p is a pattern, $Count_c(i,p)$ is the number of cases when i and c match as arguments of p in the corpus related to category c, $Count_c(x)$ is the total number of matches of x in the corpus related to category c, $TruInst$ is a set of trusted instances, $TruPat$ is a set of trusted patterns, and (t) is an iteration.

Instances and patterns with higher support are considered to be trusted. To define trusted patterns and istances, we use FILTERBYTHRESHOLD procedure, which is implemented in two versions using two different strategies.

The first strategy uses a certain threshold on *Support* value that is computed after the first iteration for patterns and instances separately. On the first iteration, the filter equals to zero, that means we allow pattern and instance extraction without any limitations. Then the threshold is set as a minimum value of support for all extracted patterns and instances correspondingly. On the next iterations, only the instances and patterns that have *Support* value greater or equal than these thresholds are added to the trusted sets. Note that within this strategy, *Support* value of any pattern and instance does not decrease. We will refer to it as THRESHOLD-SUPPORT. This is the main strategy for CPL-RUS.

THRESHOLD-SUPPORT does not limit trusted elements during algorithm run. It is greedy in sense that it collects all possible instances and patterns that are trusted enough and use them to extract new patterns and instances. Thus, final filtering should be applied in this case after the algorithm stops and the final instances, which has support not less than a certain *minimal support*, should be selected.

The second strategy uses a threshold on a number of elements of the trusted sets. After extracting new instances and patterns, they are sorted with respect to their *Support*, and then 50 most reliable instances and patterns are left in the trusted sets. We assume that this procedure would be able to correct errors made on the earlier iterations, when the algorithm have more evidence. This strategy was used in (Duarte and Hruschka, 2014). We will refer to it as THRESHOLD-50.

3.3 Implementation

Our implementation of CPL component is summarized in Algorithm 1. The algorithm processes each category c separately. It starts with a set of

[4] This particular example would probably produce the same error on the English translation, though we believe that such cases should be more rare. Since English has almost no morphology some other mechanism should be used to restrict over-production of patterns; in particular, distinguishing between verb subject and object is easier for a free-word-order language.

Algorithm 1 COUPLED PATTERN LEARNER (CPL-RUS).

Require: set of trusted patterns $\mathrm{TruPat}_c^{(0)}$, set of trusted instances $\mathrm{TruInst}_c^{(0)}$, text corpus T_c
Ensure: $\mathrm{Pat}_c^{(\infty)}$, $\mathrm{Inst}_c^{(\infty)}$
  $t \leftarrow 0$
  repeat
    $\mathrm{CandInst} \leftarrow \textsc{Extract}(\mathrm{TruPat}_c^{(t)})$
    $\mathrm{TruInst}_c^{(t+1)} \leftarrow \mathrm{TruInst}_c^{(t)} \cup \mathrm{CandInst}$
    $\textsc{FilterByThreshold}(\mathrm{TruInst}_c^{(t+1)})$
    $\mathrm{CandPat} \leftarrow \textsc{Extract}(\mathrm{TruInst}_c^{(t)})$
    $\mathrm{TruPat}_c^{(t+1)} \leftarrow \mathrm{TruPat}_c^{(t)} \cup \mathrm{CandPat}$
    $\textsc{FilterByThreshold}(\mathrm{TruPat}_c^{(t+1)})$
    $t \leftarrow t + 1$
  until $\mathrm{TruInst}_c^{(t+1)} \setminus \mathrm{TruInst}_c^{(t)} \cup \mathrm{TruPat}_c^{(t+1)} \setminus \mathrm{TruPat}_c^{(t)} = \emptyset$

trusted patterns, $\mathrm{TruPat}_c^{(0)}$, a set of trusted instances, $\mathrm{TruInst}_c^{(0)}$, and a preprocessed corpus for each c: we use only sentences that contains c lexeme(s) to speed up iterations.

Though this algorithm should run infinitely with more and more data (that is how the original NELL process organized), only small corpora are used in our experiments, and the process stops if no more patterns or instances are found during the previous iteration.

4 Experiments

4.1 Data

We use Russian Wikipedia as the data source due to the convenience of downloading a relatively small corpus devoted to some particular topic (e.g. animals) using Wikipedia categories.[5] However, we do not use a specific Wikipedia structure for anything but corpus collection, thus our method can work with any other source types. Note, that even though the Wikipedia format for articles has its own standards, all of them are written by different people with changing of author style across documents. That makes Wikipedia a good resource to obtain way the data with some varieties in style.

We use Petscan service[6] to download Wikipedia pages that belong to a certain category. For initial experiments, we collect several corpora trying to select wide but not too general categories. For example, we consider *animals* to be too general and split it into several subcategories, such as *birds*, *fish*, etc. The rational is that too broad categories might be too computationally heavy for initial experiments, while too narrow categories might not contain enough data. In total, we use a corpus of 2.5 million sentences extracted from 7 various categories (see Table 4.1). Then we annotate text with morphological attributes, such as part-of-speech, case, number, and lexeme, using Pymorphy tool (Korobov, 2015).

Wikipedia category	Number of pages	Ontology category
ANIMALS	32,412	BIRD FISH MAMMAL REPTILE
COUNTRIES	305,217	COUNTRIES
FOOD	6204	PRODUCTS
VEGETABLES	523	VEGETABLES
FRUITS	329	FRUITS
PRODUCTS	5580	FOOD
SPORT	136,027	SPORT

Table 1: Downloaded Wikipedia pages for CPL input corpus.

The results of the processing are lists of extracted patterns and instances for each category.

4.2 Initial Ontology

The initial ontology consists of 9 categories and 41 instances; it is presented in Table 4.2.

Note that FRUIT and VEGETABLE are subcategories for FOOD; we run all three independently that allow us to compare the algorithm performance on more general vs. more narrow categories.

The seed CPL patterns and their morphological constraints are listed in Table 4.2.

4.3 Experiment Design

We run experiments for all categories independently. Then we collect all extracted instances and manually annotate them as correct or incorrect. Then for each category c, we evaluated precision using the following formula:

$$Precision(c) = \frac{CorrInst(c)}{AllInst(c)},$$

[5] Wikipedia categories are different from those in ontology though they can be easily matched.

[6] https://petscan.wmflabs.org/

Category	Initial instances
BIRD	Robin, blackbird, cardinal, oriole
FISH	Shark, anchovy, bass, haddock, salmon
MAMMAL	Bear, cat, dog, horse, cow
REPTILE	Alligator, chameleon, snake, turtle
GEOGRAPHY	Africa, Canada, Brazil, Iraq, Russia
SPORT	Football, basketball, tennis
FOOD	Pepper, ice, biscuit, cheese, apple
FRUIT	Orange, peach, lemon, kiwi, pineapple
VEGETABLE	Cucumber, tomato, carrot, turnip, celery

Table 2: Seed ontology for Russian CPL (English translation).

Pattern	Arg1, case	Arg2, case	Arg1, num	Arg2, num	Arg1, pos	Arg2, pos
arg1, такие как arg2 arg1, such as arg2	nomn	nomn	plur	all	noun	noun
arg2 являются arg1 arg2 is arg1	ablt	nonm	all	all	noun	noun
arg2 относятся к arg1 arg2 refer to arg1	datv	nomn	all	all	adjf	noun
arg2 относятся к arg1 arg2 refer to arg1	datv	nomn	all	all	noun	noun

Table 3: Initial trusted patterns for Russian CPL for all categories (English translation).

where $CorrInst(c)$ is the number of correct instances extracted for category c, and $AllInst(c)$ is the whole number of instances, that were extracted by CPL for category c.

When we use the THRESHOLD-SUPPORT strategy, we perform a final filtering using different minimal support values. For algorithm comparison, we use values 0.1, 0.5 and 1.0

The main experiment is devoted to CPL-RUS with THRESHOLD-SUPPORT strategy. The algorithm converges after 6–10 iterations depending on category. We run it on all the categories and investigate the dependency of precision on support value used to cut off trusted instances after the algorithm converges.

In addition, we perform a set of smaller experiments to study CPL properties and impact of different parameters. We test: 1) usefulness of morphological features; 2) usefulness of pattern seeds; 3) differences between threshold selection strategies.

In the first experiment, we compare CPL-RUS and a version of this algorithm which do not use morphology (thus, similar to the English CPL). We will refer to the second one as CPL-NOMORPH. We run it on three ontology categories: VEGETABLE, FRUIT, and FOOD. The first run uses morphological constraints and the second allows words in all morphological forms.

In the second experiment, we investigate if the usage of seed patterns can improve the quality of the algorithm; the same experiment was conducted by (Duarte and Hruschka, 2014). As can be seen from the description in Section 2, CPL can learn without seed patterns, relying only on the set of initial categories and instances. However, since the initial ontology is small, this might be not the optimal strategy. We will refer to the second algorithm as CPL-NOPAT. We run the algorithms on the same three categories: VEGETABLE, FRUIT, and FOOD.

In the third experiment, we compare two *Threshold* selection strategies described in Section 3.3: THRESHOLD-SUPPORT, based on minimal *Support* after the first iteration and THRESHOLD-50 that keeps the fixed number of patterns and instances and revise the trusted lists after each iteration.

5 Results and Discussion

5.1 On CPL-RUS

Table 5.1 shows the main results of running CPL-RUS on the whole ontology using seeds.

There is a huge variety in results among categories with COUNTRY and SPORT being the most problematic ones despite the minimum support. FOOD as the more general category performs much worse than more narrow VEGETABLE and FRUIT, though for these categories the number of extracted instances is very low (see Table 5.2).

Interestingly, CPL-RUS with minimal support 0.5 shows better results in terms of precision than with minimal support 1. It means that some false positives have a very high $Support$ value.

5.2 On Morphological Constraints

The results of evaluating the importance of including morphological constraints to the Russian CPL are shown in Table 5.2. The precision for all categories, in this case, is much lower, which makes CPL-NOMORPH completely useless. While CPL-RUS can achieve precision 1.0 for VEGETABLE and FRUIT categories, the maximum result for the same categories in unconstrained mode is 0.43.

Table 5.2 presents results on comparison of the learning progress for the three categories with and without morphological constraints. As can be seen, morphological constraints decrease the number of extracted instances and patterns and slow down the training process.

5.3 On Usage of Seed Patterns

Table 5.3 shows the results for running CPL-NOPAT, which does not use any seed patterns. In comparison with CPL-RUS (Table 5.1), this algorithm yields worse precision, especially for the more general FOOD category. Table 5.3 shows the total number of extracted instances in both cases. As can be seen, running algorithm without seed patterns increases its coverage but decreases the resulting precision.

5.4 On Threshold Selection Strategies

Precision for different thresholds of $Support$ in CPL-RUS is shown in Figure 2. The numerical values of precision for three minimal support values are shown in Table 5.1.

In our final experiment, we test THRESHOLD-50 strategy that re-arrange patterns and instances on every step and allows only 50 of them to be trusted. The results for four ontology categories are shown in Table 5.4. Precision is better for that strategy, but the number of extracted instances is very small. It means that this strategy yields lower Recall (which is hard to evaluate in exact numbers). This gives us the opportunities for future work to find the way to determine the minimal support value that would satisfy both conditions: the number of extracted instances should not be small, and the precision should be high and does not vary among categories.[7]

5.5 Comparison with Other Approaches

The results of our experiments can be compared with the two previous work on this approach in English and Portuguese languages. Because in this work we extend the basic CPL algorithm only with morphological features of the Russian language, it makes it easy to compare the accuracy of our CPL realizations. The average accuracy for the English CPL version of the algorithm is reported as *0.78* with the minimum as *0.2* for the SPORTS EQUIPMENT category and maximum as *1.0* for the ACTOR, CELEBRITY, FURNITURE and SPORTS LEAGUE categories (Carlson et al., 2010a). The maximum average accuracy for the Russian language is *0.612*. As it can be seen, the results for the Russian language also vary between different categories, from *0.16* to *1.0*, but the average algorithm accuracy is higher for the English language. The results for the Portuguese version of CPL are presented separately for 5, 10, 15, 20 iterations of the algorithm (Duarte and Hruschka, 2014). Since we did not run more than 10 iterations of CPL for each category, the most valuable result of comparison of two CPL realizations is to choose the accuracy of 10-iterations of the Portuguese CPL. The results of the average accuracy for the Portuguese CPL is varied from *0.04* to *0.95* (Duarte and Hruschka, 2014).

6 Related Work

In this paper, we focus on coupled pattern and instance learning from the text for ontology learning; the papers related to this topic are briefly

[7] One of the reviewers suggested that it may be also useful to use a human-in-the-loop procedure, where a threshold is defined manually after a certain number of iterations using procedure similar to what we used for evaluation.

Category	Number of instances	Precision		
Minimal support		1	0.5	0.1
BIRD	315	**0.875**	0.828	0.707
FISH	731	0.242	0.403	**0.46**
MAMMAL	258	**0.685**	0.619	0.555
REPTILE	42	**0.833**	**0.833**	0.727
COUNTRY	1205	**0.272**	0.244	0.2
SPORT	1356	0.16	**0.17**	**0.17**
FOOD	204	**0.42**	0.41	0.323
VEGETABLE	16	**1.0**	**1.0**	0.9
FRUIT	1	**1.0**	**1.0**	**1.0**
Average		0.610	**0.612**	0.560

Table 4: Results of CPL-RUS.

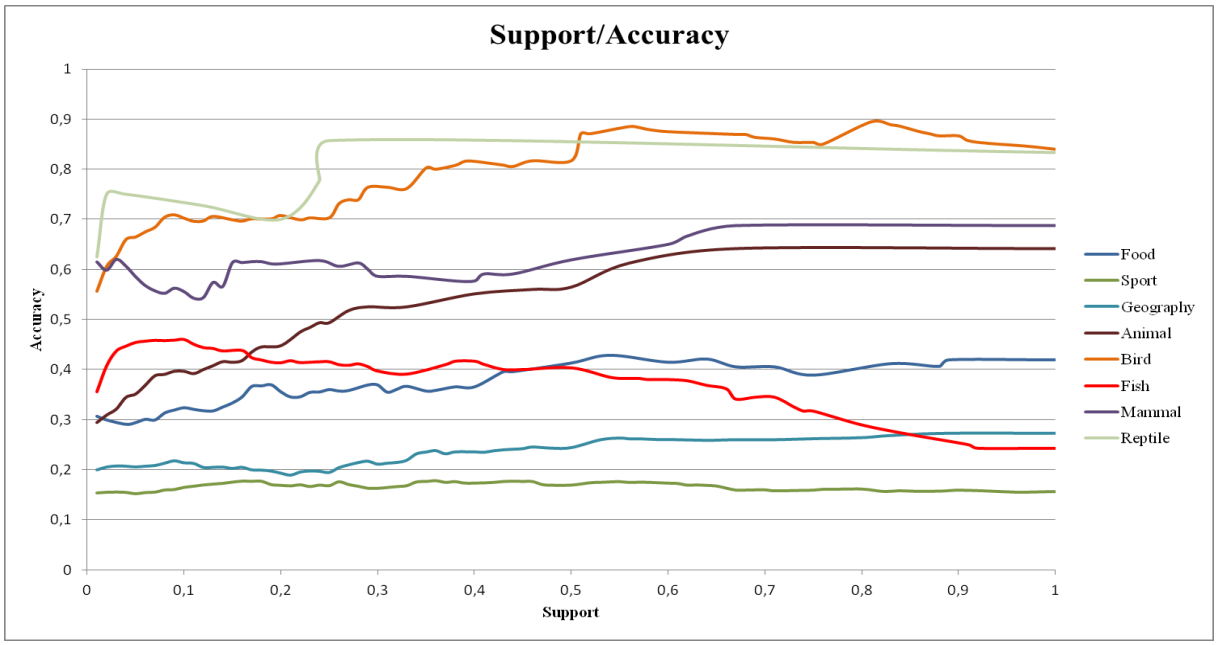

Figure 2: Dependence of CPL-RUS precision on minimal support value.

Category	Number of instances	Precision		
Minimal support		1	0.5	0.1
FOOD	1350	0.14	0.14	0.14
VEGETABLE	335	0.04	0.06	0.06
FRUIT	10	0	0	0.43

Table 5: Results of CPL-NOMORPH.

overviewed in this section. More general introduction to NELL and its predecessors can be found in (Carlson et al., 2010a).

Bootstrapping is well-known as a method for semi-supervised pattern learning. It was initially proposed for Information Extraction, that is for the traditional setting when the event templates are given beforehand (Riloff et al., 1999; Agichtein and Gravano, 2000; Yangarber, 2003). Bootstrapping for ontology learning from text has been applied, for example, by (Liu et al., 2005; Paliouras, 2005; Brewster et al., 2002).

Later the same principle was adapted for Open-Domain Information Extraction, aiming at discovering entity relations without any restrictions on their type (Shinyama and Sekine, 2006; Banko et al., 2007; Wang et al., 2011).

The idea of automatic extracting of domain templates from large corpus has been extensively studied, for example, by (Filatova et al., 2006; Chambers and Jurafsky, 2011; Fader et al., 2011). Thus, pattern-based information extraction as re-

Iteration/ Category	FRUIT		VEGETABLE		FOOD	
	inst	pat	inst	pat	inst	pat
1	0/2	10/13	0/3	42/139	2/7	37/154
2	1/3	7/10	7/158	50/548	8/416	59/2264
3	0/5	10/10	4/121	42/475	29/696	37/1227
4	0	0	1/43	9/233	39/143	78/0
5	0	0	0/9	0/87	21/63	163/0
6	0	0	0/1	0/14	17/22	213/0
7	0	0	0	0	36/3	131/0
8	0	0	0	0	26/0	72/0
9	0	0	0	0	9/0	101/0
10	0	0	0	0	13/0	53/0

Table 6: Number of extracted instances and patterns in case of using/non-using morphological constraints.

Category	Number of instances	Precision		
Minimal support		1	0.5	0.1
FOOD	262	0.07	0.09	0.17
VEGETABLE	12	0.75	0.86	0.73
FRUIT	1	1	1	1

Table 7: Results for CPL-NOPAT.

Category	with seeds	without seeds
BIRD	551	652
FISH	731	890
MAMMAL	264	267
REPTILE	45	45
COUNTRY	1204	1276
SPORT	1358	1412
FOOD	204	273
VEGETABLE	16	20

Table 8: The number of extracted instances for each category with/without seed patterns.

search field becomes closer to ontology learning and knowledge-base population, though the latter task might be more difficult since it requires cross-document inference (Ji and Grishman, 2011).

The idea of simultaneous (coupled, joint) learning of both instances and relation have been justified. Li and Ji (2014) argued that though these two tasks are traditionally broken down into separate components, this is a rather artificial division leading to over-simplification and error propagation from the earlier tasks to the later steps.

Using a knowledge base to extract relations has been previously proposed as a distant supervision approach by, among others, (Mintz et al., 2009; Surdeanu et al., 2012; Riedel et al., 2013), though

Category	Number of instances	Precision
BIRD	3	1.0
FISH	1	1.0
MAMMAL	50	0.96
REPTILE	4	0.95

Table 9: Results for running CPL-RUS with THRESHOLD-50.

these works assumed that the KB is rather big (such as Freebase).

As far as we aware, this is the first work on the application of pattern learning techniques for the Russian language, despite general interest in Information Extraction (Starostin et al., 2016) and building of linguistic resources (Loukachevitch and Dobrov, 2014; Braslavski et al., 2016). Bocharov et al. (2010) and Sabirova and Lukanin (2014) used rule-based approach to extract taxonomic relations from text. Kuznetsov et al. (2016) applied a number of machine learning techniques to automatic relation extraction from the Russian Wikipedia but their method depends on the specific structure of Wikipedia.

7 Conclusion

In this work, we made the first attempt to adapt the NELL approach to the Russian language. We changed CPL component, so it can work with morphology. We conducted several experiments with the extended version, CPL-RUS algorithm on the corpus containing over 2.5 million sentences. Our main findings are the following:

- it is possible to adapt CPL for Russian with relatively little efforts;
- the morphological constraints are crucial for Russian pattern learning;
- a small set of manually compiled seed patterns increases CPL accuracy;
- the obtained results vary for different categories; that probably means that the algorithm settings should be optimized independently for each category.

This work leaves a room for further experiments. We plan to run CPL on much bigger datasets, including the whole Wikipedia corpus and other web-pages. This would require an expansion of the seed ontology and, probably, a construction of seed patterns individually for each category or a group of categories.

We will also continue working on threshold selection strategies. Another line of research is to run CPL on top of syntactic annotation; in principle, this should increase precision though some amount of errors might be introduced by syntax parser itself.

Acknowledgments

Authors would like to thank Maisa Duarte and Estevam Hrushka for assistance during experiments preparation and for giving examples of initial ontology for CPL algorithm. This research is supported by the Government of Russian Federation, Grant 074-U01.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM.

Thomas Albertsen and Eva Blomqvist. 2007. Describing ontology applications. In *European Semantic Web Conference*, pages 549–563. Springer.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676.

Victor Bocharov, Lidia Pivovarova, Valery Rubashkin, and Boris Chuprin. 2010. Ontological parsing of encyclopedia information. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 564–579. Springer.

Pavel Braslavski, Dmitry Ustalov, Mikhail Mukhin, and Yuri Kiselev. 2016. Yarn: Spinning-in-progress. In *Proceedings of the 8th Global WordNet Conference, GWC 2016*, pages 58–65.

Christopher Brewster, Fabio Ciravegna, and Yorick Wilks. 2002. User-centred ontology learning for knowledge management. In *International Conference on Application of Natural Language to Information Systems*, pages 203–207. Springer.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and Tom M. Mitchell. 2010a. Toward an architecture for never-ending language learning. In *AAAI*, volume 5, page 3.

Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka, and Tom M. Mitchell. 2010b. Coupled semi-supervised learning for information extraction. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 101–110. ACM.

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 976–986. Association for Computational Linguistics.

Chun Wei Choo, Brian Detlor, and Don Turnbull. 2013. *Web work: Information seeking and knowledge work on the World Wide Web*. Springer Science & Business Media.

Maisa C. Duarte and Estevam R. Hruschka. 2014. How to read the web in Portuguese using the never-ending language learner's principles. In *2014 14th International Conference on Intelligent Systems Design and Applications*, pages 162–167. IEEE.

Susan Dumais, Edward Cutrell, Jonathan J. Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins. 2016. Stuff I've seen: a system for personal information retrieval and re-use. In *ACM SIGIR Forum*, volume 49, pages 28–35. ACM.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics.

Elena Filatova, Vasileios Hatzivassiloglou, and Kathleen McKeown. 2006. Automatic creation of domain templates. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 207–214. Association for Computational Linguistics.

Vladislav Grozin, Kseniya Buraya, and Natalia Gusarova. 2016. Comparison of text forum summarization depending on query type for text forums. In *Advances in Machine Learning and Signal Processing*, pages 269–279. Springer.

Thomas R. Gruber. 1995. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5):907–928.

Nicola Guarino. 1998. Formal ontology and information systems. In *Proceedings of FOIS*, volume 98, pages 81–97.

Graeme Hirst. 2009. Ontology and the lexicon. In *Handbook on ontologies*, pages 269–292. Springer.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1148–1158. Association for Computational Linguistics.

Mikhail Korobov. 2015. Morphological analyzer and generator for Russian and Ukrainian languages. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 320–332. Springer.

Artem Kuznetsov, Pavel Braslavski, and Vladimir Ivanov. 2016. Family matters: Company relations extraction from wikipedia. In *International Conference on Knowledge Engineering and the Semantic Web*, pages 81–92. Springer.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *ACL (1)*, pages 402–412.

Wei Liu, Albert Weichselbraun, Arno Scharl, and Elizabeth Chang. 2005. Semi-automatic ontology extension using spreading activation. *Journal of Universal Knowledge Management*, 1:50–58.

Natalia Loukachevitch and Boris Dobrov. 2014. Ruthes linguistic ontology vs. Russian wordnets. In *Proceedings of Global WordNet Conference GWC-2014*, pages 154–162.

Alexander Maedche. 2012. *Ontology learning for the semantic web*, volume 665. Springer Science & Business Media.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Georgios Paliouras. 2005. On the need to bootstrap ontology learning with extraction grammar learning. In *International Conference on Conceptual Structures*, pages 119–135. Springer.

Marius Pasca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. 2006. Organizing and searching the world wide web of facts-step one: the one-million fact extraction challenge. In *AAAI*, volume 6, pages 1400–1405.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. *NAACL HLT 2013*, pages 74–84.

Ellen Riloff, Rosie Jones, et al. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*, pages 474–479.

Kristina Sabirova and Artem Lukanin. 2014. Automatic extraction of hypernyms and hyponyms from Russian texts. In *Supplementary Proceedings of the 3rd International Conference on Analysis of Images, Social Networks and Texts (AIST 2014)/Ed. by DI Ignatov, MY Khachay, A. Panchenko, N. Konstantinova, R. Yavorsky, D. Ustalov*, volume 1197, pages 35–40.

Ivan Samborskii, Andrey Filchenkov, Georgiy Korneev, and Aleksandr Farseev. 2016. Person, organization, or personage: Towards user account type prediction in microblogs. In *Proceedings of First New Zealand Text Mining Workshop (TMNZ) in conjunction with the 8th Asian Conference on Machine Learning (ACML 2016)*, pages 1–13.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 304–311. Association for Computational Linguistics.

Steffen Staab and Rudi Studer. 2013. *Handbook on ontologies*. Springer Science & Business Media.

Sergey Starostin, Viktor Bocharov, Svetlana Alexeeva, Anastasiya Bodrova, Alexander Chuchunkov, Irina Efimenko, Dmitriy Granovsky, Vladimir Khoroshevsky, Irina Krylova, et al. 2016. Factrueval 2016: Evaluation of named entity recognition and fact extraction systems for Russian. In *Computational Linguistics and Intellectual Technologies. Proceedings of the Annual International Conference «Dialogue»(2016)*, number 15, pages 702–720.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465. Association for Computational Linguistics.

Wei Wang, Romaric Besançon, Olivier Ferret, and Brigitte Grau. 2011. Filtering and clustering relations for unsupervised information extraction in open domain. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1405–1414. ACM.

Roman Yangarber. 2003. Counter-training in discovery of semantic patterns. In *Proceedings of the 41st Annual Meeting on Association for Computational*

Linguistics-Volume 1, pages 343–350. Association for Computational Linguistics.